TEXAS INSTRUMENTS

Reader Series 4000

S4100 Multi-Function Reader Module   RF-MGR-MNMN

Boot Loader Reference Guide

11-06-24-701 October 2003

First Edition - October 2003

This is the first edition of this manual. It describes the **TI Series 4000 Reader.**

It contains a description of the following reader module:

S4100 Multi-Function Reader Module	P/N: **RF-MGR-MNMN-N0**

Texas Instruments (TI) reserves the right to make changes to its products or services or to discontinue any product or service at any time without notice. TI provides customer assistance in various technical areas, but does not have full access to data concerning the use and applications of customer's products

Therefore, TI assumes no liability and is not responsible for customer applications or product or software design or performance relating to systems or applications incorporating TI products. In addition, TI assumes no liability and is not responsible for infringement of patents and/or any other intellectual or industrial property rights of third parties, which may result from assistance provided by TI.

TI products are not designed, intended, authorized or warranted to be suitable for life support applications or any other life critical applications which could involve potential risk of death, personal injury or severe property or environmental damage.

The **TIRIS** logo, the words RFID Systems, **TIRIS** and are trademarks or registered trademarks of Texas Instruments Incorporated (TI).

Copyright © 2003 Texas Instruments Incorporated (TI).

This document may be downloaded onto a computer, stored and duplicated as necessary to support the use of the related TI products. Any other type of duplication, circulation or storage on data carriers in any manner not authorized by TI represents a violation of the applicable copyright laws and shall be prosecuted.

Preface

Read This First

About This Manual

This reference guide for the Series 4000 Multi-Function (13.56 MHz & 134.2 KHz) Reader is designed for use by TI customers who are engineers experienced with RFID Systems and Radio Frequency Identification Devices (RFID).

Device Name	Boot Loader Firmware Version
RF-MGR-MNMN-N0	1.02

The Regulatory, safety and warranty notices that must be followed are provided in Chapter 2.

Conventions

The following pictograms and designations are used in the operating instructions:

⚠	**WARNING:** **A WARNING IS USED WHERE CARE MUST BE TAKEN, OR A CERTAIN PROCEDURE MUST BE FOLLOWED, IN ORDER TO PREVENT INJURY OR HARM TO YOUR HEALTH.**
⚠	**CAUTION:** **This indicates information on conditions, which must be met, or a procedure, which must be followed, which if not needed could cause permanent damage to the system.**
☞	**Note:** Indicates conditions, which must be met, or procedures which must be followed, to ensure proper functioning.
ℹ	**Information:** Indicates conditions or procedures that should be followed to ensure proper functioning of the system.

If You Need Assistance

Application Centers are located in Europe, North and South America, the Far East and Australia to provide direct engineering support.

For more information, please contact your nearest TIRIS Sales and Application Center. The contact addresses can be found on our home page: http://www.tirfid.com.

Numerical Representations

Unless otherwise noted, numbers are represented as decimal.

Hexadecimal numbers are represented with the suffix $_{16}$, e.g. $A5F1_{16}$

Binary numbers are represented with the suffix $_{2}$, e.g. 1011_{2}

Byte representations: the least significant bit (lsb) is bit 0 and the most significant bit (msb) is bit 7.

Document Overview

Chapter 1

Download Tool

Topic **Page**

1.1 Introduction

This document describes the functionality of the Boot loader for the Multi Frequency Reader (MFR) Module. The Boot loader will provide a way to upgrade or replace the application that resides within the MFR Module. The Module will not have to be opened to have the application upgraded or replaced. A special software application can be made available for a PC attached to a MFR Module. This software package will send the new application to be loaded into the MFR Module.

An additional packet will be added to allow the application code to force the MFR Module into the Boot loader logic. There will be additional packets that will only work when the Module is in the Boot loader Mode. These packets are used to manipulate the Module's FLASH memory.

The potential for the application becoming corrupted increases when functionality is added to the Module while change out the application. The Module must be able to detect when the application has been removed or corrupted. The Module must also be able to recover to a state where it will be able to download a new application when the application has been corrupted or erased. The Boot loader will provide that functionality.

The user must be able to identify when a MFR Module is in the Boot loader Mode, therefore the LED's will be toggled at a special pattern to indicate when the Module is in the Boot loader Mode. This document will be focused on the initial phase of the Boot loader for the MFR Module, and is subject to change on later versions.

1.2 Programming Tools

The packet structure and its implementation are independent of the MFR's programming tools, and the tools used depend on the individual project at hand. The Boot loader has been developed with a CodeVision AVR™ compiler.

1.3 Short Description

The main purpose of this document is to outline the specifics of the Boot loader application, as it is important to understand what the Boot loader is, and how it functions. The MFR's microprocessor consists of three types of memory, the size of which may vary from chip to chip, each containing a section of FASH Memory, EEPROM, and SRAM. Both the application code and the Boot loader code will be loaded into the Flash Memory. The FLASH will be broken into a Boot loader section and a Program section.

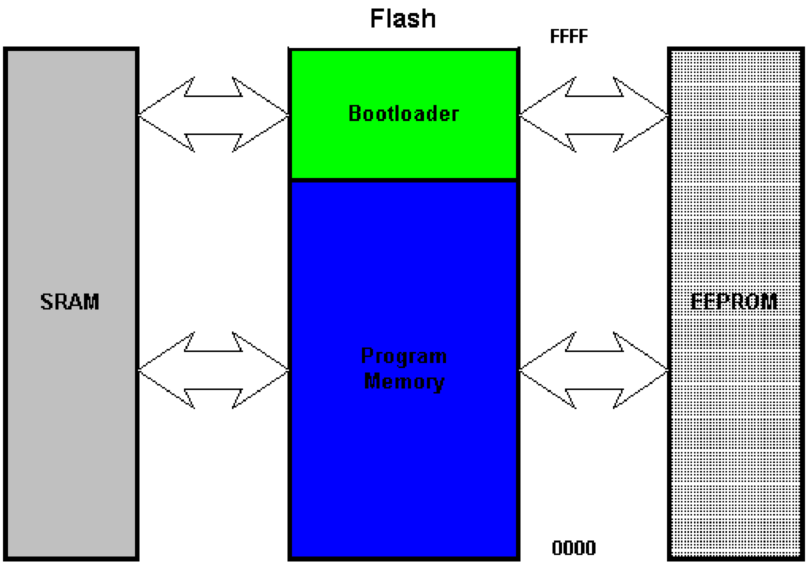


Figure 1.1 Functional Block Diagram

Both the Boot loader and Program area of FLASH will be allowed to access the SRAM and EEPROM memory but only the Boot loader section can write to FLASH. The SRAM will be used for temporary data storage and will be reset upon each power cycle.
The Boot loader will use the SRAM memory to monitor communication status and to hold the packet data before writing it to the Program area of FLASH. The EEPROM memory will not be cleared upon a power cycle so it will be used to store information such as Keys or Flags that should not be reset upon a power cycle. The application in the program area of Flash will be able to access both SRAM and EEPROM but it cannot write to FLASH memory.

Executing the Boot loader

The MFR's microprocessor provides an option of pointing to the start of the Program Area or the start of the Boot loader area on a reset or power cycle. There is a fuse bit that can be set to have the Module always execute the Boot loader first. This fuse bit will be set so that the MFR Module will always execute the Boot loader on any reset or power cycle. The main application will also provide a method of entering and executing the Boot loader.

The MFR Module must enter the Boot loader and determine if there is a valid application loaded in the program area before it attempts to go to and execute the main application. If the application has become corrupt, the Boot loader will clear out the Program area and stay in the Boot loader waiting for an application to download. The LEDs will flash in a 'heartbeat' pattern to indicate to the user that a problem has been detected and it is in the Boot loader waiting for an application download.

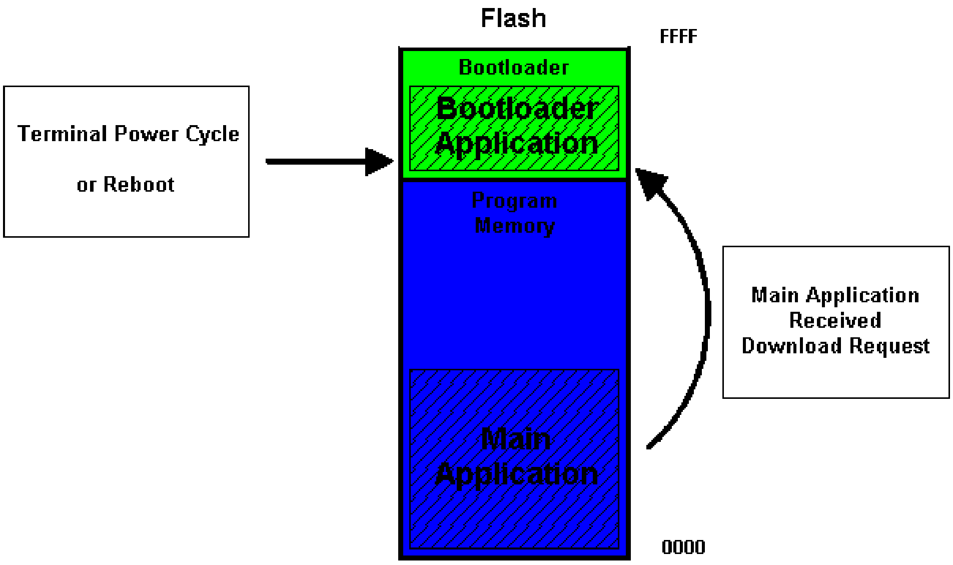


Figure 1.2 Programming Process Flow

The Boot loader will jump to the main application and execute it when all of the application verification tests pass. The verification tests will take minimal time and should not provide any obvious delays before executing the main application. A large delay may cause the user to think the MFR Module is not functioning when it is.

It is also possible that a user may want to upgrade or replace the main application in the Module. The main application will support a new POS packet that will force the MFR Module to enter and execute the Boot loader application. When the Module enters the Boot loader from the Application Mode, a timeout period will begin. The Boot loader will send a Response Packet to the Request Packet that forced the Module into the Download Mode. The MFR Module will wait for 15 seconds to receive a packet that is a valid Download Request Packet. Any other packets will be ignored until the timeout has expired and the Module has returned to the main application. This will prevent the user accidentally locking up the Module, or erasing the main application.

The timeout period will be reset when each valid Download Request Packet is received within the timeout period. When the timeout period does expire, the Boot loader will determine if it can go back to the main application or not. If all the tests pass, then the MFR Module will return to and execute the main application upon a timeout expiring or a restart. The Boot loader will erase all of the Program memory area and return to the first state of the Download Mode when the timeout expires or there is a restart and any of the tests fail. The Module will stay in the Boot loader in Download Mode until a new application has been successfully loaded and verified.

1.4 Exiting the Boot loader

The Boot loader will be required to make some intelligent decisions before it exits the Download State and executes the main application that resides in the Program area of FLASH. The Boot loader must determine when to exit Download Mode and execute the main application, after it has determined that a valid application does exist and it is not waiting for a timeout period.

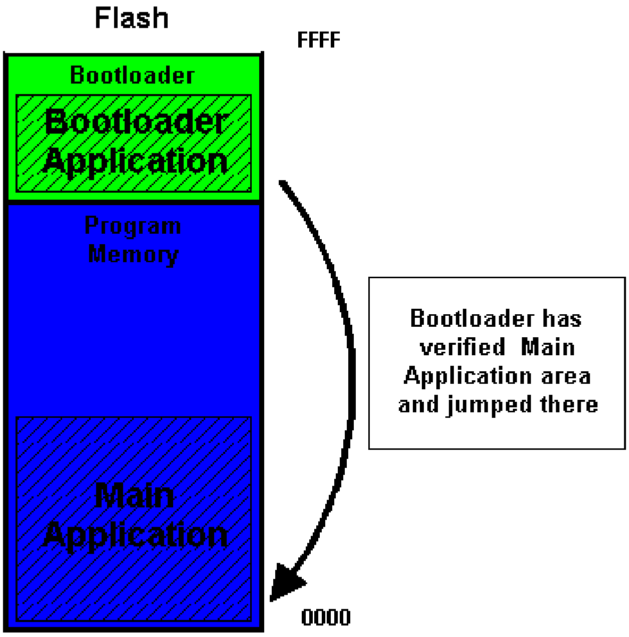


Figure 1.3 Programming Process Flow

The simplest scenario is one in which the MFR Module is simply power cycled and the Boot loader is executed. The Boot loader will be required to determine if the main application is loaded and not corrupted before jumping to and executing the main application. The objective here is to validate the Program area of Flash as fast as possible and to begin the main application. The Boot loader will simply exit by jumping to and executing the application that resides in the program area of FLASH. This validation process delay should not be obvious to an end user.

The next scenario is one in which the main application has received an Enter Download Mode Request packet and jumped into the Boot loader. This scenario would require the Boot loader to wait for a valid Download Request Packet after it validates the main application. The Boot loader will wait for 15 seconds before returning to the main application. A flag will be set in the EEPROM memory to indicate that the main application forced the Boot loader into Download Mode. The Boot loader will clear this flag and set the timeout for 15 seconds and begin sending the ‘heartbeat’ LED pattern while waiting for a valid Download Request Packet. If no valid Download Request Packet has been received within the timeout period, the MFR Module will exit the Boot loader Download Mode and return to the main application.

The final scenario involves the program area of FLASH failing the validation tests. This memory could have been cleared by the Boot loader after detecting an error, or may have been cleared after receiving a Download Request Packet requesting that the program be erased. This memory will be cleared if the Download Request Packets are received in the wrong order. The reason why the validation test failed is not important. The main application will be deleted when any of the tests fail. Once the application is gone, the Boot loader must not exit until proper sequence of Download Request Packets have been received and the main application is ready to be executed. The sequence of Download Request Packets will be covered in the next section.

One important point to understand is that you will not be able to exit the Boot loader Download Mode until the process is complete. Attempting to exit early will cause the application to be deleted and the Download process to start again. The proper sequence of Download Request Packets will load a new main application then verify the CRC and then exit the Boot loader and jump to the main application. If a person attempts to hack into the Module and insert code it will result in the Module erasing the main application and returning to the Boot loader Download Mode.

Information:

You will not be able to exit the Boot loader Download Mode until the process is complete. Attempting to exit early will cause the application to be deleted and the Download process to start again.

1.5 Download Request Packet Flow

The MFR Module may enter the Boot loader Download Mode after receiving a Download Mode Request packet or may be forced into that mode when the Boot loader detects some kind of error. Regardless of how the Module entered the state, the functionality will remain the same while in this Download State. The Module will also be sending a ‘heartbeat’ pattern to light the LEDs while monitoring the COM port waiting for receipt of a valid Download Packet. Note that the Download Packets are different than a Download Mode Request Packet. The functionality and flow of these Download Request Packets will be covered in this section.

The Module will start a 15 second timeout when it enters the Download Mode. The timeout will be reset to 15 seconds each time the Module receives a valid Download Request Packet. If this timeout period does elapse, the Module will run some tests and decide if it will jump to and execute the main application or erase the program area and restart the Download State and timeout period.

Some of the Download Request Packets must be sent in a particular order or they will result in the main application being deleted and the Download Mode being reset. This will prevent hacking into the Module and attempting to insert patches of code. There are two Download Request Packets that are basically wildcards to the Download State table or flow. These commands can be executed at any point while in the Download Mode. They will reset the timeout period and transmit a response packet, but they will not affect the Download State table or flow.

The two commands that are wild cards are the Get Boot Version Request Packet and the Init Download Request Packet. Either of these commands can be used to confirm that the Module is in the Download Mode. The Module will ignore the packets when it is not in the Download Mode. The Module will reset the 15 second timeout and return a response if it is in the timeout mode. The Get Boot Version Request will return the Boot loader version in the response packet. The Init Download request will return the status if it successfully initialized the parameters requested. Note that the device communicating to the MFR Module must be prepared for all changes if the Init Download request were to change something such as the baud rate or parity. Both devices must be at the same rate and parity. An example of this might be to enter the Download State at 9600 baud, but increase the speed while downloading the new application. It is also important to note that the Module will always return to default settings when the Download State is reset.

Information:

Note that the device communicating to the MFR Module must be prepared for all changes if the Init Download request were to change something such as the baud rate or parity. Both devices must be at the same rate and parity.

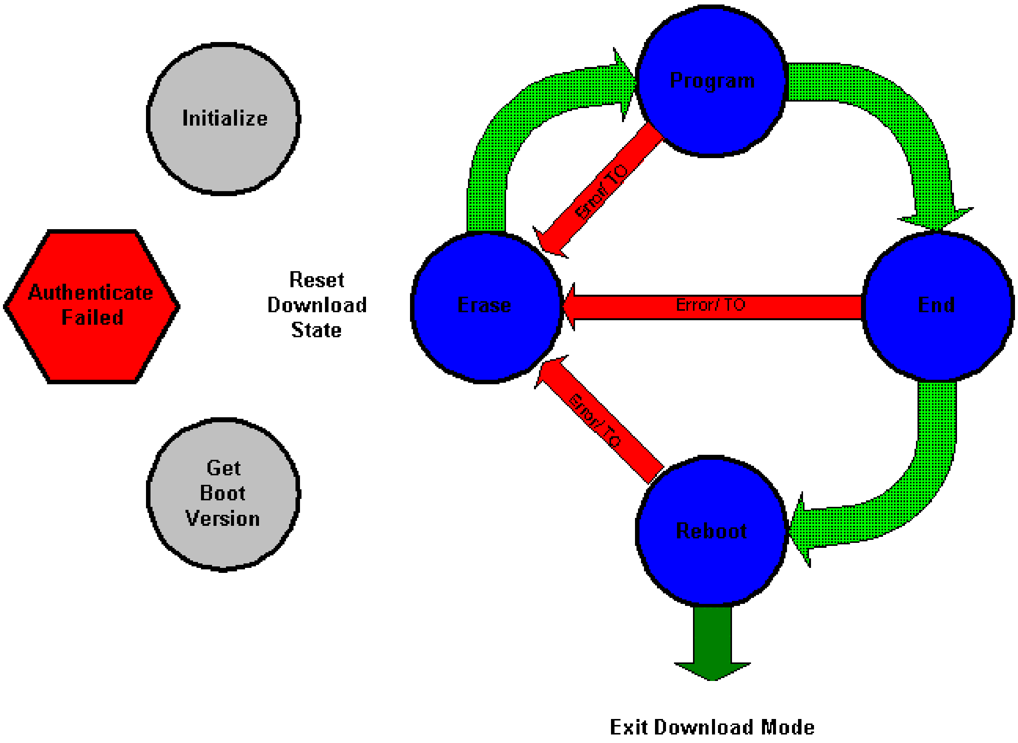


Figure 1.4 Programming Process Flow

The Erase state will be the true starting point of the Download process. This state can be reached by receiving an Erase Request Packet. The Boot loader will also place the Module into this state if it detects any problem in Main application area or the Download Request packets have been received out of sequence. Each of the other states will return to this state if they detect an error or a timeout has occurred between valid Download Request Packets. The intent is to allow the Module to detect problems and be able to recover to a predictable state and default settings. This will prevent probing and patching into the application as well as keeping the Module from locking up if the problem is due to the application communicating to the MFR Module.

The next state after the Erase state will be the Program state. This is the point where the MFR Module will receive the new main application one page at a time. The overhead from the Intel HEX format will be stripped off and sent 128 bytes at a time. The data will be written to the SRAM memory via the communication protocol and then will be read from the SRAM and written to the program area of FLASH. Each Program Request Packet will contain an address that will indicate the page number where this data will be written. The Module will respond with a response packet that will indicate if the write was successful. The program sending the packets will be responsible for and resend a page if the page was not written to memory. This will allow the pages to be sent in any order. All of the pages must be sent before doing an End or Reboot or the Module will erase the program area and restart the Download Mode. The Program state will not make a lot of intelligent decisions. The MFR Module will trust the program sending the packets to direct the data to the proper page. The MFR will not send an error if the program skips pages or rewrites pages it will just send the response packet stating if the write was successful. The program sending the data will be required to determine how many times to resend a page that is failing before halting the process. The MFR Module will detect when it has not received a valid Download Request Packet within the timeout duration and then check the program memory to see if it is valid. If this timeout occurs before all the pages have been written the result will be to return to the first Download State and erase the program memory. Note that the Module must have entered the Erase state prior to entering the Program state or it will be forced to the Erase state and clear out the main application.

The Module may enter the End state after all of the application pages have been successfully written to the program area. The Module must receive an End Download Request Packet to enter the End state. A validation process will be run on the program memory. If the test fails, the Module will erase all of the program memory and restart the Download State. If the test does pass, the Boot loader will run a CRC calculation on the entire Program Memory area and save the CRC value EEPROM and change the Download State to indicate the Module is no longer in the Download Mode.

The last Download State is the Reboot state. When the MFR Module receives a Reboot Request packet and the tests pass, it will a reboot and start at the beginning of the Boot loader code. The Module Boot loader will run a CRC calculation on the entire program memory and compare it to that stored in the EEPROM. If the two values do not match, the Boot loader will erase all of the program memory and restart the Boot loader. The Module will exit the Boot loader and jump to and execute the main application if the checksum passes and the Module is in the proper Download State.

1.6 Download State

The MFR Module will reserve a byte of EEPROM memory to represent the Download State. This byte will be used by the Boot loader to determine the functionality the Boot loader will support at that time. This state will also be used for security to make sure the Module is not being hacked into.

Download State Byte	Brief description
0	Not in Download Mode
1	Application has forced to Download Mode
2	Authentication Failed
3	Download Required
4	Memory Write

Table 1.1 Download State Byte

The Download State will be set to **0** most of the time. This will indicate that the Boot loader has not detected any CRC error in the Program memory. The timeout period for waiting for a valid Download request packet will be set to zero, so the Module will jump directly to the main application and execute it after the Module has been power-cycled.

The Download State will only be set to **1** when the main application has received a Download Mode Request Packet. The Boot loader will start the ‘heartbeat’ LED pattern and the 15-second timeout to wait for a valid Download Request Packet. The Download State will be reset to **0** and the Module will jump to and execute the main application if no valid packet is received before the timeout expires.

The Download State will be set to **3** when the Boot loader has erased the Program memory and reset the default communication parameters. The Module can enter this state by receiving an Erase Request Packet or detecting a CRC error on the Program memory. The Module will never be allowed to attempt to execute the main application while in this Download State.

The Download State will advance to a **4** as soon as the Module receives a valid Program Request Packet. Note that the Module will not be allowed to accept a Program Request Packet unless it is already in Download State **3** or **4**. The receipt of a Program Request Packet from any other Download State will result in the Module returning to Download State **3** and erasing the Program memory and resetting the communication defaults.

The Module must be in Download State **4** when it receives a valid End Request Packet. If the Module is in any other state it will return to Download State **3** and erase the Program memory and resetting the communication defaults. If the Module is in Download State **4** and it receives the request packet and the CRC test pass, the Download State will be reset to **0**. At that point the Module can access the main application after a Reboot or power-cycle. The Download State **2** will be reserved for future development.

1.7 Response Status Byte Codes

The response packets will contain a response status byte. This byte will always be the first byte of the data area of the response packet. The data following the response status byte will depend on the request packet sent, the device, the Module type, and the response status byte returned. The response status byte may be the only byte in the data area of the response packet. The Module will not return a response packet if the request packet is invalid or has been sent to a device and Module type that do not support the request packet. The following table will provide a potential list of error commands but not all error commands are applicable to each request packet. This list is provided as an example of what the status bytes could be and it is not final and will change during the development process.

Response Status byte code in hex	Description of the Response Status Byte code	Module Type
00	No error / Success	HSM, MT, TPU
01	Token Not Present	MT, TPU
02	Authentication Failed	MT, TPU
03	Read Error	MT, TPU
04	Data Integrity Error	MT, TPU
05	Invalid Payload Data Format	MT, TPU
06	RF Coupler not responding	MT, TPU
07	Mutual Authentication - Timeout	HSM
08	Mutual Authentication – Data Invalid	HSM
09	Key Injection Failed - Timeout	HSM
0A	Key Injection Failed – Data Invalid	HSM
0B	RFC Injection Error	HSM
0C	LF not responding	MT, TPU
0D	HF not responding	MT, TPU
0E	Device ID Invalid	MT, TPU
0F	MAC Failed	MT, TPU
10	Illegal Sub System ID	MT, TPU
11	Wrong Download State	MT, TPU
12	Write Failed	MT, TPU
13	Invalid Address	MT, TPU
14	Invalid Baud	MT, TPU
15	Invalid Check digits	MT, TPU
16	SC error – packet invalid/timeout	HSM, MT, TPU
17	Verify Password - data invalid	TPU
18	Load Key – data invalid	HSM
19	Module personalization – TPU/MT not found	HSM
1A	Module personalization – packet	HSM

	invalid/timeout	
1B	HSM SC reports error during Module personalization	HSM
1C	Term SC reports error during Module personalization	HSM
20	14443 – success	MT, TPU
21	14443 – write failure	MT, TPU
22	14443 – write timeout	MT, TPU
23	14443 – read timeout	MT, TPU
24	14443 – ATQB:error processing ATQB	MT, TPU
25	14443 – ATQB:AID invalid	MT, TPU
26	14443 – ATQB:Number of applications field invalid	MT, TPU
27	14443 – ATQB: Bit rata capability error	MT, TPU
28	14443 – ATQB: Max frame size error	MT, TPU
29	14443 – ATQB: Protocol type invalid	MT, TPU
2A	14443 – ATTRIB: Invalid MBLI	MT, TPU
2B	14443 – Not an I-block (I-block expected)	MT, TPU
2C	14443 – Invalid CID	MT, TPU
2D	14443 – Token does not have sufficient power	MT, TPU
2E	14443 – Not a R-block (R-block expected)	MT, TPU
2F	14443 – Not a S-block (S-block expected)	MT, TPU
30	14443 – Could not initialize SLF9000N FIFO	MT, TPU
31	14443 – ATQB: ADC/FO error	MT, TPU
32	14443 – Handle S(WTXM) block (not an error code)	MT, TPU

Table 1.2 Response Status Byte Codes

1.8 Download Request Packet Flow Chart

The Module will Power up in the Boot loader Application. The Module will verify that there is a valid Target Application before exiting the Boot loader Application and running the Target Application. If the Boot loader detects any Problem it will Erase the Target Application and enter Download State **3**.

The Module can also enter the Boot loader Application when the Target Application receives an Enter Download Mode Request. The Download Request Packet flow is shown in the following diagram.

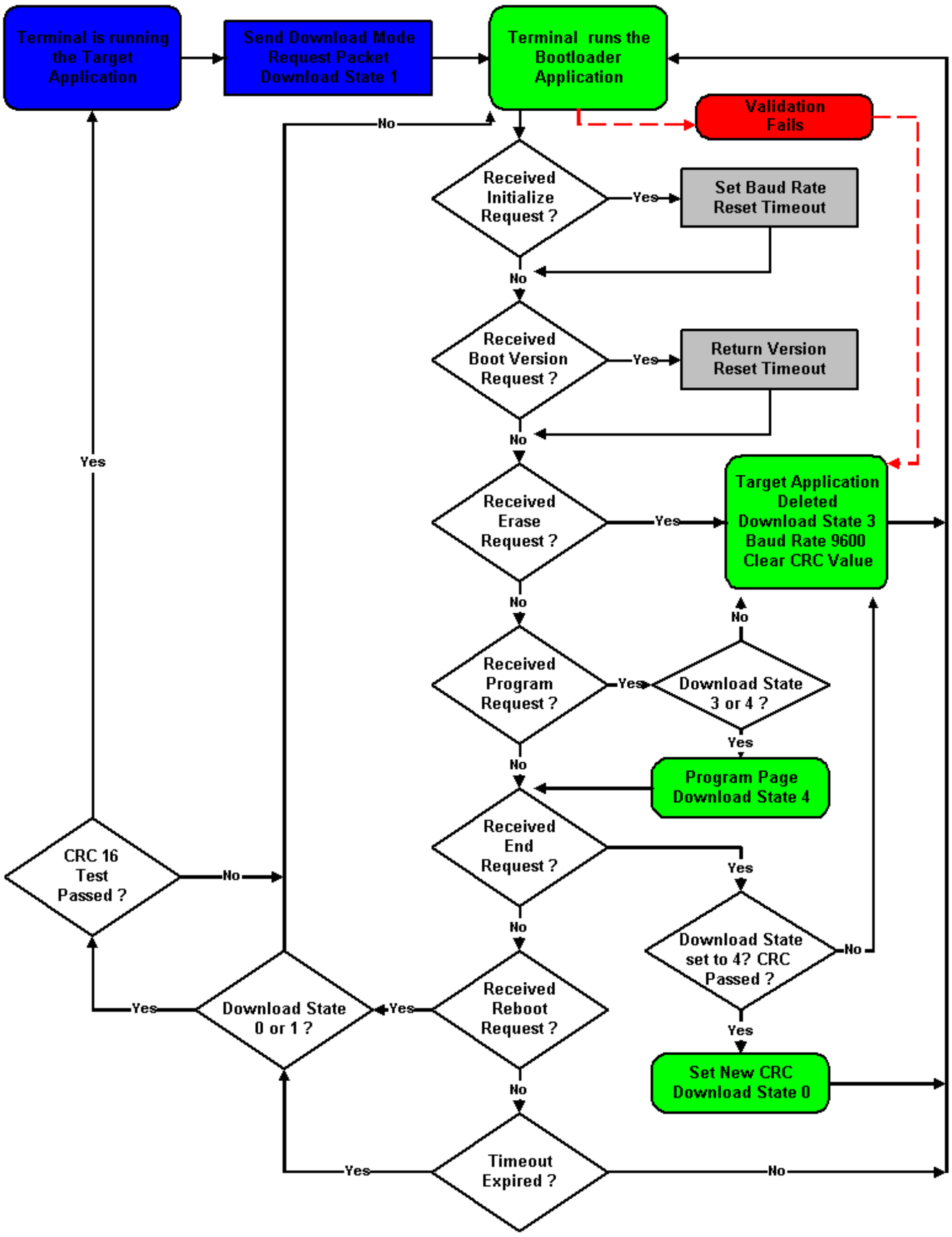
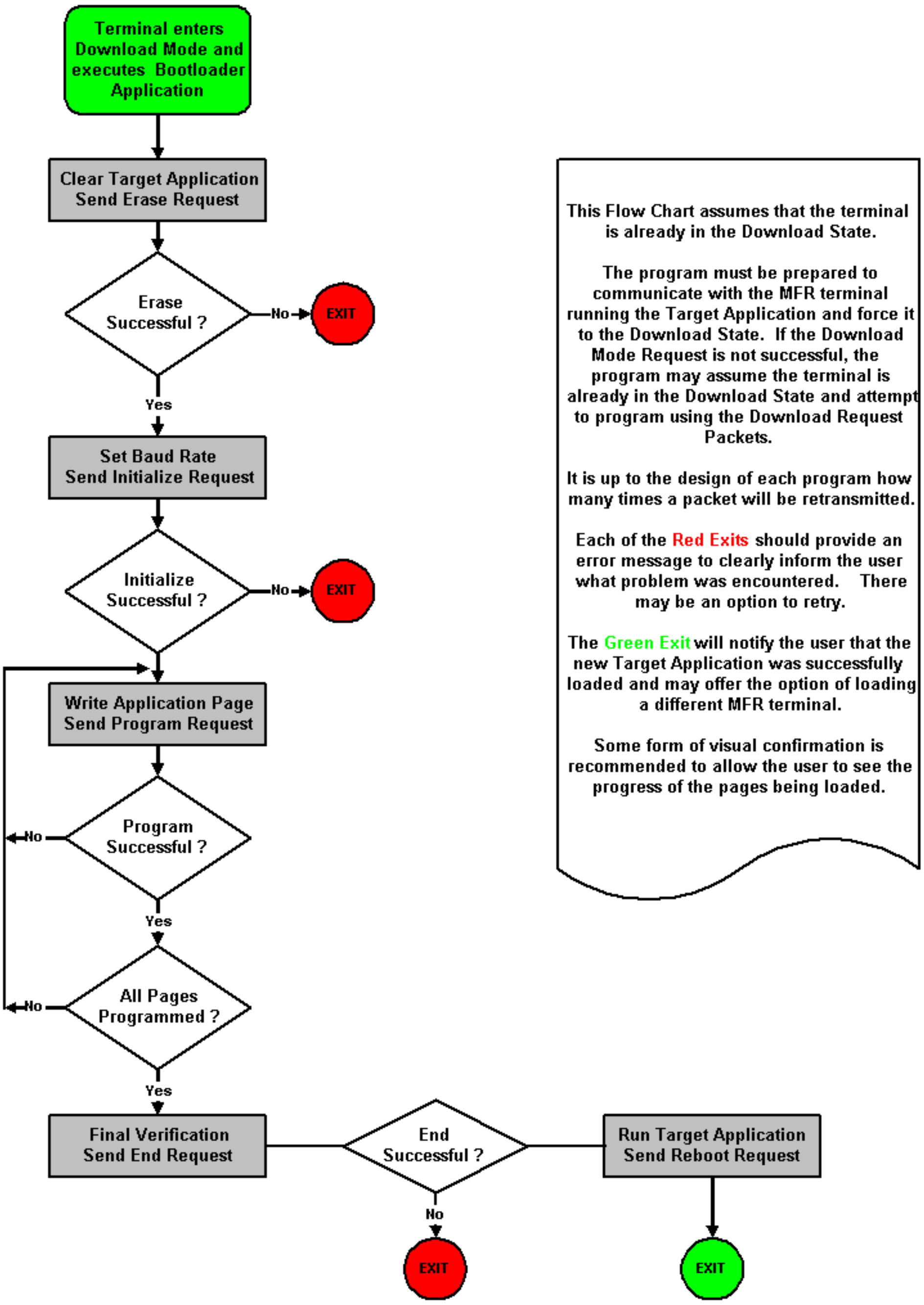

Terminal is running the Target Application
Send Download Mode Request Packet Download State 1
Terminal runs the Bootloader Application
Validation Fails
No
Received Initialize Request ?
Yes
Set Baud Rate Reset Timeout
No
Received Boot Version Request ?
Yes
Return Version Reset Timeout
No
Yes
Received Erase Request ?
Yes
Target Application Deleted Download State 3 Baud Rate 9600 Clear CRC Value
No
No
Received Program Request ?
Yes
Download State 3 or 4 ?
Yes
No
Program Page Download State 4
Received End Request ?
Yes
CRC 16 Test Passed ?
No
No
Download State set to 4? CRC Passed ?
No
Yes
Download State 0 or 1 ?
Yes
Received Reboot Request ?
Yes
Set New CRC Download State 0
No
Yes
Timeout Expired ?
No

Figure 1.5 Download Request Packet Flow

1.9 Download Tool Packet Flow

TI may provide a tool that will take a Hex file and pass it to the Module in a format that will allow the Boot loader Application to load it into the Program area of FLASH and then run it as the Target Application. There are countless options in how a programmer may choose to design this tool. The following flow chart shows the recommended Download Request Packet Flow.



This Flow Chart assumes that the terminal is already in the Download State.

The program must be prepared to communicate with the MFR terminal running the Target Application and force it to the Download State. If the Download Mode Request is not successful, the program may assume the terminal is already in the Download State and attempt to program using the Download Request Packets.

It is up to the design of each program how many times a packet will be retransmitted.

Each of the Red Exits should provide an error message to clearly inform the user what problem was encountered. There may be an option to retry.

The Green Exit will notify the user that the new Target Application was successfully loaded and may offer the option of loading a different MFR terminal.

Some form of visual confirmation is recommended to allow the user to see the progress of the pages being loaded.

Figure 1.9 Download Tool Packet Flow

Chapter 2

Regulatory and Warranty Notices

Topic **Page**

2.1 FCC Conformity

The Series 4000 Multi-Function Reader is an intentional radiator. The transmitter portion operates at 13.56 MHz and is subject to FCC Part 15, Subpart C, “Intentional Radiator,” paragraph 15.225 (13.553-13.567MHz). Radiated emissions from the device are subject to the limits in Section 15.209 of the Rules outside of the 13.56 +/- 0.007 MHz band.

Note:

Any device or system incorporating the Series 4000 reader, in full or in part, needs to obtain FCC certification as part of the system within which this reader unit resides. A system containing this product may be operated only under an experimental license or final approval issued by the relevant approval authority. Before any such device or system can be marketed, an equipment authorization must be obtained form the relevant approval authority.

2.2 ETSI Conformity

Any device or system incorporating the Series 4000 reader, in full or in part, may need to comply with European Standard EN300330. It is the responsibility of each system integrator to have their complete system tested and to obtain approvals as required from the local authorities before operating or selling this system.

2.3 CE Conformity

Any device or system incorporating the Series 4000 reader, in full or in part, may need to have a CE Declaration of Conformity stating that it meets European EMC directive 99/5/EC. This must be issued by the system integrator or user of such a system prior to marketing or operating it in the European community.

2.4 Warranty and Liability

The "General Conditions of Sale and Delivery" of Texas Instruments Incorporated or a TI subsidiary apply. Warranty and liability claims for defect products, injuries to persons and property damages are void if they are the result of one or more of the following causes:

- Improper use of the reader module.
- Unauthorized assembly, operation and maintenance of the reader module.
- Operation of the reader modules with defective and/or non-functioning safety and protective equipment.
- Failure to observe the instructions during transport, storage, assembly, operation, maintenance and setting up of the reader modules.
- Unauthorized changes to the reader modules.
- Insufficient monitoring of the reader modules' operation or environmental conditions.
- Improperly conducted repairs.
- Catastrophes caused by foreign bodies and acts of God.